(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 686 910 A1**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **95109062.0**

(22) Date of filing : **12.06.95**

(51) Int. Cl.^{6} : **G06F 7/48,** G06F 9/32

(30) Priority : **10.06.94 JP 151799/94**

(43) Date of publication of application :
13.12.95 Bulletin 95/50

(84) Designated Contracting States :
DE FR GB

(71) Applicant : **NEC CORPORATION**
7-1, Shiba 5-chome
Minato-ku
Tokyo (JP)

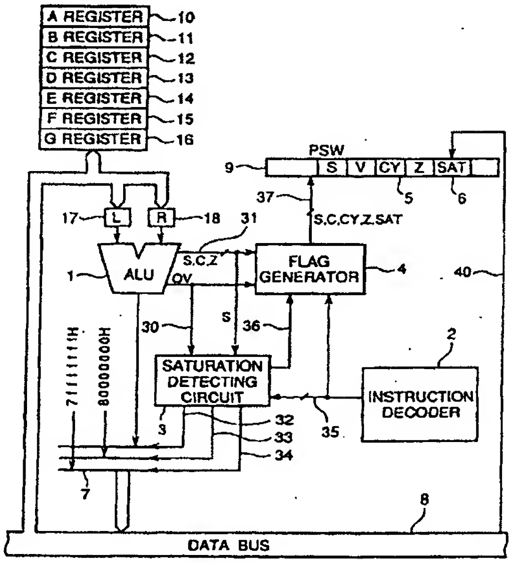
(72) Inventor : **Okayama, Sachiko**
c/o NEC Corporation,
7-1, Shiba 5-chome
Minato-ku, Tokyo (JP)
Inventor : **Katsuta, Hiroshi**
c/o NEC Corporation,
7-1, Shiba 5-chome
Minato-ku, Tokyo (JP)

(74) Representative : **Baronetzky, Klaus, Dipl.-Ing.**
et al
Patentanwälte
Dipl.-Ing. R. Splanemann, Dr. B. Reitzner,
Dipl.-Ing. K. Baronetzky
Tal 13
D-80331 München (DE)

(54) Data processing system having a saturation arithmetic operation function

(57) A data processing system includes an instruction decoder for decoding a string of instructions including an arithmetic operation instruction, an arithmetic operation unit controlled by the instruction decoder for executing a designated arithmetic operation for a received data, the arithmetic operation unit outputting not only the result of the designated arithmetic operation, but also a sign information and an overflow/underflow information of the result of the designated arithmetic operation, and a saturation detecting circuit receiving the sign information and the overflow/underflow information for controlling a selcctor in such a manner that if an overflow has occurred when the sign information indicates the positive, the selector selects a positive maximum value ; if an underflow has occurred when the sign information indicates the negative, the selector selects a negative maximum value ; and if neither the overflow nor the underflow has occurred, the selector selects the result of arithmetic operation outputted from the arithmetic operation unit. Furthermore, if the overflow has occurred when the sign information indicates the positive, or if the underflow has occurred when the sign information indicates the negative, the saturation detecting circuit sets a saturation flag of a PSW indicative of existence/nonexistence of a saturation processing. After the saturation flag is set, the saturation flag is reset by only execution of a data transfer instruction by the arithmetic operation unit.

FIGURE 4



EP 0 686 910 A1

Jouve, 18, rue Saint-Denis, 75001 PARIS

Background of the Invention

Field of the invention

The present invention relates to a data processing system, and more specifically to a data processing system having an arithmetic operation function for performing a modification processing after an overflow.

Description of related art

A saturation processing is a processing for determining a maximum value or a minimum value which is a correct sign as the result of the arithmetic operation, when an overflow or an underflow occurs in an arithmetic system mainly in the process of executing an cumulative summing and subtracting operation.

When an integer is expressed by a binary number in the form of a 2'compliment, for example, a 32-bit data having a sign information (sign bit) constituted of the most significant bit (MSB), can indicate a numerical value in a range from a negative minimum value "-1" expressed by FFFFFFFFH (suffix "H" indicates a hexadecimal notation in this specification) to a negative maximum value expressed by 80000000H, and in another range from from a positive minimum value "0" expressed by 00000000H to a positive maximum value expressed by 7FFFFFFFH.

When the sign bit is "1", the data indicates a negative integer, and when the sign information is "0", the data indicates a positive integer.

When the sign bit is located at the MSB, for example, if the positive integer which is the result of the arithmetic operation overflows over the positive maximum value, at the moment a carry occurs from "7FFFFFFFH", the sign bit will change from "0" to "1". As a result, the result of the arithmetic operation which should be expressed as a positive integer, is expressed as a negative integer.

To the contrary, if the negative integer which is the result of the arithmetic operation underflows over the negative maximum value, at the moment a borrow occurs from "80000000H", the sign bit will change from "1" to "0". As a result, the result of the arithmetic operation which should be expressed as a negative integer, is expressed as a positive integer.

The result of the arithmetic operation which had caused a sign inversion as mentioned above, cannot be used for control, since it is inverted in sign from a correct result of arithmetic operation.

Therefore, when the result of the arithmetic operation exceeds the positive or negative maximum value, it has been so set that the result of the arithmetic operation becomes the maximum value in a correct sign. This has been called to "saturate" or a "saturation processing". In addition, of arithmetic operation instructions, an instruction capable of performing the saturation processing is called as a "saturation arithmetic operation instruction".

In the prior art, for example, Japanese Patent Application Laid-open Publication No. JP-A-58-115544 proposed one exarnple of an arithmetic system having a saturation arithmetic operation function for performing the saturation processing.

This proposed arithmetic system is configured to set an overflow flag and to modify the result of arithmetic operation to a positive or negative maximum value, when the result of arithmetic operation causes an overflow or an underflow.

Furthermore, there has been also proposed to provide a saturation arithmetic operation instruction for performing the saturation processing and an ordinary arithmetic operation instruction which does not perform the saturation processing, and to set a C bit (indicative or a carry, a borrow, an overflow, an underflow or a shift-out) and to modify the result of arithmetic operation to a positive or negative maximum value, if an overflow or an underflow.occurs when the saturation arithmetic operation instruction has been executed. In this case, when the ordinary arithmetic operation instruction is executed, if an overflow or an underflow.occurs, the C bit is set, but the result of arithmetic operation is maintained as it is.

Now, a construction and an operation of the conventional data processing system will be described with reference to the drawings.

Referring to Figure 1, there is shown a typical example of the conventional arithmetic system having a saturation arithmetic operation instruction dedicated for performing the saturation processing. In Figure 1, Reference Numeral 1 designates a 32-bit arithmetic and logic unit, and Reference Numeral 2 shows an instruction decoder for decoding various instructions (not shown) including the saturation arithmetic operation instruction, the ordinary arithmetic operation instruction, and others and for outputting an arithmetic operation command 35. Reference Numeral 3 indicates a saturation detecting circuit for detecting that the result of arithmetic operation is saturated, and Reference Numeral shows a flag generator for setting the result of the instruction execution to a PSW (program status word) 9.

Furthermore, Reference Numeral 5 designates a CY flag indicating that a carry or a borrow has occurred in the result of the arithmetic operation, and Reference Numeral 7 shows a selector for selecting one of the result of arithmetic operation, a positive maximum value and a negative maximum value. In addition, Reference Numeral 9 is the PSW for recording the result of arithmetic operation in the ALU 1 and the status of the CPU, and Reference Numeral 8 is a data bus. Reference Numerals 10 to 16 designate an A register, a B register, a C register, a D register, an E register, an F register and a G register, respectively, which are a 32-bit registers and which constitute a general purpose register for storing various data used for a program execution. Reference Numeral 17 shows a L (left) register for giving one input of the ALU 1, and Reference Numeral 18 indicates a R (right) register for giving the other input of the ALU 1.

In accordance with the arithmetic operation command 35, the ALU 1 executes a designated arithmetic operation such as an addition, a subtraction, a multiplication and a division for data held in the L register 17 and data held in the R register 18. The ALU outputs the result of arithmetic operation to the selector 7, and also supplies the flag generator 4 with a status signal 30 (including information OV indicative of whether or not the result of arithmetic operation causes the overflow or the underflow), and another status signal 31 (including a sign information S of the result of arithmetic operation, a carry/borrow information C and information Z indicating whether or not the result of arithmetic operation is "0" (zero)).

The status signal 30 and the sign information S of the status signal 31 outputted from the ALU 1 are supplied to the saturation detecting circuit 3.

The instruction decoder 2 outputs the arithmetic operation command 35 to the saturation detecting circuit 3 and the flag generator 4. The arithmetic operation command 35 outputted to the flag generator 4 is composed of a signal activated when the instruction decoder 2 decodes the saturation arithmetic operation instruction.

The saturation detecting circuit 3 outputs to the flag generator 4 a status signal indicating that the saturation processing has been perfomed. In addition, the saturation detecting circuit 3 selectively activates selection signals 32, 33 and 34 for selecting the result of arithmetic operation.

On the basis of the selection signals 32, 33 and 34, the selector 7 selects one of the result of arithmetic operation outputted from the ALU 1, a positive maximum value and a negative maximum value, and outputs the selected one to the data bus 8.

On the basis of the status signals 30 and 31, the flag generator 4 outputs to the PSW 9 various flag control signals 37 for changing the status of a corresponding flag only when an instruction allowing a change of the corresponding flag is executed.

The A register 10 to the G register 16 supply data through the data bus 8 to the L register 17 and the R register 18. These L register 17 and R register 18 supply data to the ALU 1.

Now, a detailed construction of the PSW 9 and the flag generator 4 will be described with reference to Figure 2. In Figure 2, elements corresponding to those shown in Figure 1 are given the same Reference Numerals.

In Figure 2, of the status signal 31 outputted from the ALU 1 shown in Figure 1, a status signal corresponding to the sign information S is expressed by "31-S", and a status signal corresponding to the carry/borrow signal C is expressed by "31-C". Furthermore, a status signal corresponding to the information Z indicating whether or not the result of arithmetic operation is "0" is expressed by "31-Z". In addition, of the arithmetic operation command 35 generated by the instruction decoder 2, a signal supplied to the flag generator 4 is expressed by "35-SAT". Of the flag control signals 37 generated by the flag generator 4, a signal supplied to the CY flag 5 is indicated as a control signal 37-CY.

Each flag included in the PSW 9 is constituted of a D-type flipflop, which is controlled to hold data at an ending timing of each instruction.

The flag control signals 37 are supplied to an data input D of corresponding D-type flipflops, each of which constitutes one flag. Explaining the control signal 37-CY and the D-type flipflop constituting the CY flag 5, the control signal 37-CY outputted by the flag generator 4 is supplied to the data input D of the D-type flipflop constituting the CY flag 5.

The flag generator 4 includes a 2-input AND circuit 41 for generating a logical product (conjunction) between the status signal 36 indicating that the saturation processing has been executed and the arithmetic operation command signal 35-SAT, another 2-input AND circuit 42 for generating a logical product between an output of the AND circuit 41 and the status signal 30, an inverter 45 generating an inverted signal of the arithmetic operation command signal 35-SAT, still another AND circuit 44 for generating a logical product between an output of the inverter 45 and the signal 31-C, and a 2-input OR circuit 43 for generating a logical sum (disjunction) between an output of the AND circuit 44 and an output of the AND circuit 44.

The status signal 36 is supplied to one input of the AND circuit 41, and the arithmetic operation command signal 35-SAT is supplied to the other input of the AND 41 and an input of the inverter 45. The signal 31-C and the output of the inverter 45 are supplied to a pair of inputs of the AND circuit 44. The output of the AND circuit

41 is connected to one input of the AND circuit 42. The status signal 30 is connected to the data input D of the D-type flipflop constituting the V flag in the PSW 9, and also to the other input of the AND circuit 42. The signal 31-S and the signal 31-Z are connected to respective data inputs D of the flipflops constituting the S flag and the Z flag in the PSW 9, respectively. An output of the OR circuit 43 is connected, as the signal 37-CY, to the data input D of the D-type flipflop constituting the CY flag in the PSW 9.

Now, with reference to Figure 1, explanation will be made on operation when the saturation arithmetic operation instruction is executed.

First, operation when the saturation processing is performed at the time of executing the saturation arithmetic operation instruction, will be described.

Under the arithmetic operation command 35 from the instruction decoder 2, data held in registers selected from the general purpose registers including the A register 10 to the G register 16 is set in the L register 17 and the L register 18. The saturation detecting circuit 3 receives the status signals 30 and 31 from the ALU 1, and also controls the selector 7 by the selection signals 32, 33 and 34 so that in response to the overflow or the underflow, the selector 7 selects either the positive maximum value or the negative maximum value in accordance with a combination of logical values of the information OV and the sign information S.

More specifically, when there occurs the overflow resulting in the sign information S of "0" and the information OV of "1", the saturation detecting circuit 3 activates the selection signal 34 so that the selector 7 selects and outputs "7FFFFFFFH" to the data bus 8. In addition, the saturation detecting circuit 3 activates the status signal 36 indicating that the saturation processing has been performed. In addition, the flag generator 4 receives the status signal 36, and outputs the flag control signal 37 for indicating that the saturation processing has been performed, so that the CY flag 5 in the PSW 9 is set to "1".

Similarly, when there occurs the underflow resulting in the sign information S of "1" and the information OV of "1", the saturation detecting circuit 3 activates the selection signal 33 so that the selector 7 selects and outputs "80000000H" to the data bus 8 In addition, the saturation detecting circuit 3 activates the status signal 36 indicating that the saturation processing has been performed, and, the flag generator 4 receives the status signal 36, and outputs the flag control signal 37 for indicating that the saturation processing has been performed, so that the CY flag 5 in the PSW 9 is set to "1".

Furthermore, referring to Figure 2, a detailed operation of the PSW 9 and the flag generator 4 when the saturation processing is performed at the time of executing the saturation arithmetic operation instruction, will be described.

At the time of executing the saturation arithmetic operation instruction, the arithmetic operation command signal 35-SAT outputted from the instruction decoder 2, is activated to "1" which indicates that the instruction is the saturation arithmetic operation instruction. Since both of the status signal 36 and the signal 35-SAT are activated to "1", the output of the AND circuit 41 is brought to "1". In addition, the output of the inverter 45 is brought to "0", so that the output of the AND circuit 44 is brought to "0" regardless of the logical level of the signal 31-C. Since the status signal 30 of the information OV is "1", the output of the AND circuit 42 is brought to "1". Accordingly, the output 37-CY of the OR circuit 43 is brought to "1", so that the CY flag 5 is set to "1".

Referring to Figure 1, again, operation when neither the overflow nor the under flow occurs at the time of executing the saturation arithmetic operation instruction, namely, in the case of the information OV of "0", will be described. In this situation, the saturation processing is not performed, so that the result of arithmetic operation is outputted through the selector 7 to the data bus 8.

Namely, the saturation circuit 3 activates the signal 32 to cause the selector 7 to output the result of arithmetic operation in the ALU 7. Since the saturation processing is not performed, the status signal 36 is inactivated, so that the flag generator 4 outputs the flag control signal 37 so as to clear the CY flag 5 in the PSW 9 to "0" in order to indicate that the saturation processing has not been performed.

Furthermore, referring to Figure 2, a detailed operation of the PSW 9 and the flag generator 4 when the saturation processing is not performed at the time of executing the saturation arithmetic operation instruction, will be described.

When the saturation arithmetic operation instruction is executed, the arithmetic operation command signal 35-SAT outputted from the instruction decoder 2, is activated to "1" which indicates that the instruction is the saturation arithmetic operation instruction. Since the saturation processing is not performed, the status signal 36 is inactivated to "0". Accordingly, although the signal 35-SAT is active, the output of the AND circuit 41 is at "0". In addition, the output of the inverter 45 for inverting the arithmetic operation command signal 35-SAT, is brought to "0", so that the output of the AND circuit 44 is brought to "0" regardless of the logical level of the signal 31-C. Since the output of the AND circuit 41 is at "0", the output of the AND circuit 42 is brought to "1" regardless of the logical level of the status signal 30. Accordingly, the output 37-CY of the OR circuit 43 is brought to "0", so that the CY flag 5 is set to "0".

The following TABLE 1 indicates a relation between the selection by the saturation detecting circuit 3 and the status of the CY flag 5. As seen from TABLE 1, the CY flag 5 is not set to "1" in response to generation of the carry or borrow. Namely, the CY flag 5 is not set to "1" on the basis of the fact that the saturation processing has been performed.

TABLE 1

SELECTION BY SATURATION DETECTING CIRCUIT 3 AND STATUS OF CY FLAG 5				
RESULT OF ARITHMETIC OPERATION	FROM ALU 1		SELECTION BY SATURATION DETECTING CIRCUIT 3	STATUS OF CY FLAG 5
	S	OV		
EXCEEDS POSITIVE MAX VALUE	0	1	7FFFFFFFH	1
EXCEEDS NEGATIVE MAX VALUE	1	1	80000000H	1
THE OTHERS	0/1	0	RESULT OF ARITHMETIC OPERATION IN ALU 1	0

Again referring to Figure 1, an operation when the ordinary arithmetic operation instruction with no saturation processing is performed, will be explained.

In the ordinary arithmetic operation instruction with no saturation processing, the saturation detecting circuit 2 is controlled to activate the signal 32 so that selector 7 always outputs the result of arithmetic operation of the ALU 1. Since no saturation processing is performed, the status signal 36 is always inactive, and on the other band, when the carry or the borrow occurs in the result of arithmetic operation, the flag generator 4 outputs the flag control signal 37 to set the CY flag 5 in the PSW 9 to "1". If neither the carry nor the borrow occurs, the flag generator 4 clears the CY flag 5 to "0".

Referring to Figure 2, again, a detailed operation of the PSW 9 and the flag generator 4 when the ordinary arithmetic operation instruction with no saturation processing is not performed, will be described.

Since the saturation arithmetic operation instruction is not executed, both of the status signal 36 and the arithmetic operation command signal 35-SAT are inactivated to "0". Therefore, the output of the AND circuit 41 is at "0", and the output of the inverter 45 is at "1", so that the logical level of the signal 31-C becomes the output of the AND circuit 44 without modification. In addition, since the output of the AND circuit 41 is at "0", the output of the AND circuit 42 is brought to "0" regardless of the logical level of the status signal 30. Accordingly, the output 37-CY of the OR circuit 43 becomes equal to the logical level of the signal 31-C, so that the CY flag 5 in the PSW 9 is either set to "1".or cleared to "0" in response to the existence/non-existence of the carry or borrow in the result of arithmetic operation of the ALU 1.

The typical example of the conventional arithmetic system capable of performing the saturation processing has been constructed as mentioned above. Namely, if the saturation processing has been performed, the CY flag 5 is set to "1", so that just after the saturation processing has been performed, by investigating the CY flag 5 in a software manner, it is possible to know that the saturation processing has been performed.

However, if the CY flag 5 is not investigated just after the saturation processing has been performed, the CY flag 5 may be cleared to "0" in the execution of its succeeding arithmetic operation This is considered that, because the CY flag (which reflects one after another and in real time the result of arithmetic operations in the ALU 1 when ordinary arithmetic operation instructions are executed) is used as a flag showing the saturation processing, even if the fact that the saturation processing has been performed is stored in the CY flag 5, the storing will extinguished by an ordinary arithmetic operation instruction executed after the saturation arithmetic operation instruction.

In the following, this will be explained in detail with reference to a specific example of a cumulative summing in a predictive filter in a system for setting the revolution number of an electric motor.

In such a system, when the revolution number of the motor is elevated to a desired revolution number, the revolution number is set while predicting a future revolution number. For example, the future revolution number is predicted by a cumulative summing expressed for example in the following equation (1)

$$Vn + 1 = \alpha * Vn + \beta * Vn - 1 + \gamma * Vn - 2 \quad (1)$$

where Vn is a current revolution number; Vn-1 is the revolution number before one period; Vn-2 is the revolution number before two periods.

The revolution number Vn+1 in a next period is predicted by multiplying the revolution numbers in the past by constants α, β and γ and summing the multiplied revolution numbers.

Here, the constants α, β and γ are set to be able to be discriminated on the basis of whether or not the result of arithmetic operation based on real parameters in each item exceeds the integer arithmetic operation range of -2^{23} to $+2^{23}$ in the 32-bit ALU 1.

If a difference between the calculated revolution number Vn+1 and the desired revolution number is "0" (zero), the setting of the revolution number is completed. If there is a difference between the calculated revolution number Vn+1 and the desired revolution number, a timing of a driving pulse for controlling the revolution number of the motor is changed. By repeatedly obtaining the difference between the calculated revolution number Vn+1 and the desired revolution number, the setting of the revolution number is maintained until the difference between the calculated revolution number Vn+1 and the desired revolution number becomes "0" (zero).

In the above mentioned step of calculating the revolution number by the cumulative summing, if the result of the summing in each item in the process of the cumulative summing is frequently saturated, it is considered to be because of deterioration with age of the motor and/or other causes, and the constants α, β and γ in the respective items are modified.

In this case, since the conventional arithmetic system is so configured that when the saturation processing has been performed, the CY flag is set to "1" and the saturation processing has not been performed, the CY flag is set to "0", if the status of the CY flag is not memorized just after the saturation processing has been performed, there is a fear that the CY flag is cleared in a next arithmetic operation. Because of this, generation of the saturation in the course of the arithmetic operation may not be detected.

Here, operation of the saturation processing will be described with reference to Figure 3 which is a flow chart illustrating the saturation processing in the conventional arithmetic operation.

Here, it is assumed that the constants α, β and γ are previously stored in the A register 10, the B register 11 and the C register 12 in the arithmetic system shown in Figure 1, respectively, and the revolution numbers Vn, Vn-1 and Vn-2 are also previously stored in the D register 13, the E register 14 and the F register 15, respectively. In addition, the G register 16 is used for store the fact that the saturation processing has been performed.

First, for initialization, the G register 16 is set to "0" (Step 200). And, α * Vn is calculated (Step 201) and β * Vn-1 is calculated (Step 202).

By the saturation arithmetic operation instruction, a sum of the values obtained in the Steps 201 and 202, namely, α * Vn + β * Vn-1 is calculated (Step 203).

Then, whether or not the saturation processing has been performed for the result of arithmetic operation in Step 203, is discriminated by referring to the CY flag 5 (Step 204). If the CY flag 5 is "1", "1" is set to the G register so as to indicate that the saturation processing has been performed (Step 205).

Thereafter, γ * Vn-2 is calculated (Step 206), and a sum of the values obtained in the Steps 203 and 206, namely, α * Vn + β * Vn-1 + γ * Vn-2 is calculated (Step 207).

Furthermore, whether or not the saturation processing has been performed for the result of arithmetic operation in Step 207, is discriminated by referring to the CY flag 5 (Step 208).

If the CY flag 5 is "1", "1" is set to the G register so as to indicate that the saturation processing has been performed (Step 209).

The above mentioned processing shown in Figure 3 in the case that an actual operation is executed using the general purpose registers in the arithmetic system shown in Figure 1, and the saturation processing is performed in the Step 203 and the processing in Step 207 is not saturated, will be now explained. Incidentally, Figure 3 additionally shows the status of the CY flag 5 and the status of the G register 16 which are changed with the progress of the processing.

In Step 200, the G register 16 is set "0", and in Step 201, a product between the A register 10 and the D register 13 is stored in the D register 13. In Step 202, a product between the B register 11 and the E register 14 is stored in the E register 14.

In Step 203, a sum of the D register 11 and the E register 14 is calculated, and in this case, the saturation processing is performed and its result is stored in the E register 14. Accordingly, the CY flag 5 is set to "1".

In Step 204, the CY flag 5 is discriminated. Since the CY flag 5 is at "1", the G register 16 is set to "1" in Step 205.

Thereafter, in Step 206, a product between the C register 12 and the F register 15 is stored in the F register 15.

In Step 207, a sum of the E register 14 and the F register 15 is calculated, and stored in the F register 15 without performing the saturation processing. In this case, accordingly, the CY flag 5 is reset to "0".

In Step 208, the CY flag 5 is discriminated. Since it is discriminated that the CY flag 5 is at "0", the processing flow is ended.

In the above mentioned example, since the result of arithmetic was saturated in Step 203, the CY flag 5 was set to "1", but since the result of arithmetic was not saturated in Step 207, the CY flag 5 was brought to "0". At this time, the information of the saturation processing in the CY flag 5 is invalidated.

On the other hand, since the fact that the result of arithmetic was saturated in Step 203, is memorized in the G register 16 in the step 205, the information indicating that the saturation processing has been performed, is held regardless of whether or not the saturation processing has been performed after the step 203.

As mentioned above, the conventional arithmetic system is so configured that when the positive result of arithmetic operation in the ALU 1 causes an overflow, namely exceeds "7FFFFFFFH", the saturation processing is performed, and thereafter, the CY flag 5 is set to "1", and on the other hand, when the result of arithmetic operation is equal to or less than "7FFFFFFFH", the CY flag 5 is reset to "0".

Therefore, in order to discriminate whether or not the result of arithmetic operation of the predicted revolution number has been saturated, by checking the CY flag 5, there are required the processing for discriminating whether or not the saturation processing has been performed, just after each saturation arithmetic operation, as shown in Step 204 and Step 208 in Figure 3, and the processing for memorizing the fact that the saturation processing has been performed, as in Step 205 and in Step 209.

As seen from Figure 3, in setting the revolution number of the motor which is rotating at a high speed, of ten steps in total of the processing used for controlling a limit value, four steps are an overhead expended for the discrimination of the saturation processing and the memorization. This gives an adverse influence on realizing a high precision and a high speed of the revolution number control. In addition, in order to memorize whether or not the saturation processing has been performed, it has been required to use the general purpose register of the arithmetic system.

Summary of the Invention

Accordingly, it is an object of the present invention to provide an arithmetic system which has overcome the above mentioned defect of the conventional one.

Another object of the present invention is to provide an arithmetic system having a saturation processing function, but making unnecessary the overhead including discriminating whether or not the saturation processing has been performed just after each saturation arithmetic operation instruction, and memorizing the fact that the saturation processing has been performed.

The above and other objects of the present invention are achieved in accordance with the present invention by a data processing system including an arithmetic operation means and a memory means including at least one flag indicating the status of the result of arithmetic operation performed in the arithmetic operation means, the memory means including at least a second flag for storing existence/non-existence of the saturation processing, the second flag being reset by execution of a predetermined instruction for resetting the second flag.

In one embodiment, after the second flag for storing existence/non-existence of the saturation processing, is set in response to execution of the saturation processing, and is maintained in a set condition until the predetermined instruction for resetting the second flag is executed.

Alternatively, the data processing system in accordance with the present invention further includes a detecting means receiving information indicative of execution of a saturation arithmetic operation instruction and a signal indicative of existence/non-existence of an overflow and another signal indicative of a sign, both outputted from the arithmetic operation means, the detecting means detecting the saturation processing, so that when the detecting means detects the saturation processing, the' second flag for storing existence/non-existence of the saturation processing, is set.

Preferably, the second flag for storing existence/non-existence of the saturation processing, is formed of a RS flipflop having a reset input activated by execution of the predetermined instruction.

With the above mentioned arrangement, when an overflow has occurred and the result of arithmetic operation is positive, or when an underflow has occurred and the result of arithmetic operation is negative, the second flag indicative of existence/non-existence of a saturation processing is set. After the saturation flag is set, the saturation flag is reset by only execution of a data transfer instruction by the arithmetic operation unit. Accordingly, the overhead for discriminating whether or not the saturation processing has been performed just before the execution of the saturation arithmetic operation instruction, and then for memorizing the fact that the saturation processing has been performed, becomes unnecessary. Thus, it is possible to realize a high precision and a high speed of the data processing system. In addition, it is no longer necessary to use the general purpose register of the arithmetic system in order to memorize whether or not the saturation processing has been performed.

The above and other objects, features and advantages of the present invention will be apparent from the following description of preferred embodiments of the invention with reference to the accompanying drawings.

Brief Description of the Drawings

Figure 1 is a block diagram of a conventional data processing system having a saturation processing function;
Figure 2 is a logic diagram of the PSW and the flag generator used in the conventional data processing system shown in Figure 1;
Figure 3 is a flow chart illustrating the saturation processing in the conventional arithmetic operation shown in Figure 1;
Figure 4 is a block diagram of an embodiment of the data processing system in accordance with the present invention having a saturation processing function;
Figure 5 is a logic diagram of the PSW and the flag generator used in the data processing system shown in Figure 4; and
Figure 6 is a flow chart illustrating the saturation processing in the arithmetic operation shown in Figure 4.

Description of the Preferred embodiments

Referring to Figure 4, there is shown a block diagram of an embodiment of the data processing system in accordance with the present invention having a saturation processing function. Figure 4, element similar to those shown in Figure 1 are given the same Reference Numerals, and explanation thereof will be omitted so that only features of the data processing system shown in Figure 4 different from the conventional one shown in Figure 1 will be described for simplification of the description.

In the embodiment shown in Figure 4, when a saturation processing has been performed in accordance with a saturation arithmetic operation instruction, the flag generator 4 sets a SAT (saturation) flag 6 in the PSW 9 to "1", in place of the CY flag 5.

The SAT flag 6 is connected to a data line 40 which is one bit of the data bus 8, so that the SAT flag 6 is cleared to "0" by execution of an instruction for transferring "0" to the SAT flag 6 under a control of a controller (not shown) of the data processing system.

Now, a detailed construction of the PSW 9 and the flag generator 4 of the shown embodiment will be described with reference to Figure 5. In Figure 5, elements corresponding to those shown in Figure 2 are given the same Reference Numerals, and explanation thereof will be omitted.

As shown in Figure 5, the PSW 9 of the shown embodiment newly provides the SAT flag 6 for holding the existence/non-existence of the saturation, and the SAT flag 6 is constituted of a RS (reset-set type) flipflop. A set terminal S of the RS flipflop constituting the SAT flag 6 is connected to receive the signal 37-SAT supplied from the flag generator 4. When the signal 37-SAT is at "1", the RS flipflop 6 is set to "1".

The data line 40 is connected to a reset input R of the RS flipflop 6 as a clear signal for the SAT flag 6. When the clear signal 40 for the SAT flag 6 is at 1", the RS flipflop 6 is reset to "0".

As shown in Figure 3, furthermore, the flag generator 4 includes only the AND circuits 41 and 42 of the various logic circuits included in the conventional flag generator shown in Figure 2. The output of the AND circuit 42 is connected to the SAT flag 6 as the signal 37-SAT, and the status signal 31-C is supplied as it is to the CY flag 5 of the PSW 9.

Operation when the saturation arithmetic operation instruction is executed, will be described with reference to Figure 5.

First, a detailed operation of the PSW 9 and the flag generator 4 when the saturation processing is perfomed at the time of executing the saturation arithmetic operation instruction, will be described.

At the time of executing the saturation arithmetic operation instruction, the arithmetic operation command signal 35-SAT outputted from the instruction decoder 2, is activated to "1" which indicates that the instruction is the saturation arithmetic operation instruction. Since both of the status signal 36 indicating that the saturation processing has been performed, and the signal 35-SAT are active, namely "1", the output of the AND circuit 41 is brought to "1". Since the status signal 30 of the information OV is "1", the control signal 37-SAT is brought to "1", so that the SAT flag 6 is set to "1".

Next, a detailed operation of the PSW 9 and the flag generator 4 when the saturation processing has not been preformed at the time of executing the saturation arithmetic operation instruction, will be described. When the saturation arithmetic operation instruction is executed, the arithmetic operation command signal 35-SAT is activated to "1". On the other hand, Since the saturation processing is not performed, the status signal 36 is inactivated to "0", so that the output of the AND circuit 41 becomes "0". Since the one input of the AND circuit 42 is at "0", the control signal 37-SAT outputted from the AND circuit 42 is brought to "0" regardless of the logical level of the status signal 30 (information OV), so that the SAT flag 6 continues to maintain its previous information.

The following TABLE 2 shows a relation between the selection by the saturation detecting circuit 3 and the status of the SAT flag 6.

TABLE 2

SELECTION BY SATURATION DETECTING CIRCUIT 3 AND STATUS OF SAT FLAG 6				
RESULT OF ARITHMETIC OPERATION	FROM ALU 1		SELECTION BY SATURATION DETECTING CIRCUIT 3	STATUS OF SAT FLAG 6
	S	OV		
EXCEEDS POSITIVE MAX VALUE	0	1	7FFFFFFFH	1
EXCEEDS NEGATIVE MAX VALUE	1	1	80000000H	1
THE OTHERS	0/1	0	RESULT OF ARITHMETIC OPERATION IN ALU 1	NOT CHANGE

In the data processing system shown in Figure 4, the operation when an ordinary arithmetic operation instruction with no saturation processing is executed, is substantially the same as the corresponding operation of the conventional example explained hereinbefore.

Referring to Figure 5, again, a detailed operation of the PSW 9 and the flag generator 4 when the ordinary arithmetic operation instruction with no saturation processing is not performed, will be described.

Since the saturation arithmetic operation instruction is not executed, both of the status signal 36 and the arithmetic operation command signal 35-SAT are inactivated to "0". Therefore, the output of the AND circuit 41 is at "0". Since the output of the AND circuit 41 is at "0", the control signal 37-SAT is brought to "0" regardless of the logical level of the status signal 30. Accordingly, the SAT flag 6 continues to maintain its previous information.

Now, explanation will be made on a case that the shown embodiment is used in the predicting filter, similarly to the conventional example explained hereinbefore.

Referring to Figure 6, there is shown a flow chart illustrating the saturation processing in the arithmetic operation in the shown embodiment of the data processing system. With reference to Figure 6, a process until the result of the cumulative summing is obtained, similarly to the conventional example, will be described.

Here, it is assumed that the constants α, β and γ are previously stored in the A register 10, the B register 11 and the C register 12 in the arithmetic system shown in Figure 4, respectively, and the revolution numbers Vn, Vn-1 and Vn-2 are also previously stored in the D register 13, the E register 14 and the F register 15, respectively. In the shown embodiment, however, the G register 16 is not used.

First, for initialization, the SAT flag 6 is set to "0" (Step 100). And, α * Vn is calculated (Step 101) and β * Vn-1 is calculated (Step 102).

By the saturation arithmetic operation instruction, a sum of the values obtained in the Steps 101 and 102, namely, α * Vn+-β * Vn-1 is calculated (Step 103). Furthermore, γ * Vn-2 is calculated (Step 104). By the saturation arithmetic operation instruction, a sum of the values obtained in the Steps 103 and 104, namely, α * Vn + β * Vn-1 + γ * Vn-2 is calculated (Step 105).

Here, the above mentioned processing shown in Figure 6 in the case that an actual operation is executed using the general purpose registers in the arithmetic system shown in Figure 4, and the saturation processing is performed in the Step 103 and the processing in Step 105 is not saturated, will be explained. Incidentally, Figure 6 additionally shows the status of the SAT flag 6 which is changed with the progress of the processing.

In Step 100, the SAT flag 6 is set "0", and in Step 101, a product between the A register 10 and the D register 13 is stored in the D register 13. In Step 102, a product between the B register 11 and the E register 14 is stored in the E register 14.

In Step 103, a sum of the D register 11 and the E register 14 is calculated, and in this case, the saturation processing is performed and its result is stored in the E register 14. At this time, the SAT flag 6 is set to "1".

In Step 104, a product between the C register 12 and the F register 15 is stored in the F register 15.

In Step 105, a sum of the E register 14 and the F register 15 is calculated, and stored in the F register 15 without performing the saturation processing. At this time, the SAT flag 6 remains in the set condition of "1". Namely,.the SAT flag 6 does not change, and the processing flow is ended.

In the above mentioned embodiment, the fact that the saturation processing has been performed in Step 103, is stored by setting the SAT flag 6 to "1", and in the case that the saturation processing is not performed in its succeeding processing, the SAT flag 6 is not cleared, and therefore, the information indicative of the fact that the saturation processing was performed, is held.

As seen from the above, in setting the revolution number of the motor which is rotating at a high speed similarly to the situation of the conventional example shown in Figure 3, the steps expended for the discrimination of the saturation processing and the memorization (which were required in the conventional example) are completely omitted from the steps of the processing used for controlling a limit value. In other words, the ten steps in total required in the conventional example are reduced to six steps. Accordingly, the overhead which occupied about 40% of the total processing time in the conventional example, is removed by only adding the very simple flag function, and it is possible to realize a high precision and a high speed of the revolution number control.

In the above mentioned embodiment, after the SAT flag for storing whether or not the saturation processing has been performed, is set once, the SAT flag is in no way cleared by the result of succeeding arithmetic operations, until a predetermined instruction is executed to reset the SAT flag. Therefore, it is sufficient if the discrimination as to whether or not the saturation processing was performed, is carried out at an end of all cumulative summing operations

In addition, in order to memorize whether or not the saturation processing has been performed, the shown embodiment does not require to use the general purpose register of the arithmetic system. In other words, the G register 16 used for memorizing whether or not the saturation processing has been performed in the conventional example explained hereinbefore, can be used for another purpose, for example, for storing another data. Therefore, the hardware resource of the data processing system can be effectively utilized.

The invention has thus been shown and described with reference to the specific embodiment in which the present invention was applied to cumulative summing for the control of the revolution number of the motor has been described. However, it should be noted that the present invention is in no way limited to the details of the illustrated structures but changes and modifications may be made within the scope of the appended claims. If the present invention is applied to various applications, it is possible to reduce the overhead in the various applications.

Claims

1. A data processing system including an arithmetic operation means and a memory means including at least one flag indicating the status of the result of arithmetic operation performed in said arithmetic operation means, said memory means including at least a second flag for storing existence/non-existence of the saturation processing, said second flag being reset by execution of a predetermined instruction for resetting said second flag.

2. A data processing system claimed in Claim 1 wherein after said second flag for storing existence/non-existence of the saturation processing, is set in response to execution of the saturation processing, and is maintained in a set condition until said predetermined instruction for resetting said second flag is executed.

3. A data processing system claimed in Claim 1 further including a detecting means receiving information indicative of execution of a saturation arithmetic operation instruction and a signal indicative of existence/non-existence of an overflow and another signal indicative of a sign, both outputted from said arithmetic operation means, said detecting means detecting the saturation processing, so that when said detecting means detects the saturation processing, said second flag for storing existence/non-existence of the saturation processing, is set.

4. A data processing system claimed in Claim 1 wherein said second flag for storing existence/non-existence of the saturation processing, is formed of a RS flipflop having a reset input activated by execution of said predetermined instruction.

5. A data processing system including an instruction decoder for decoding a string of instructions including an arithmetic operation instruction, an arithmetic operation unit controlled by said instruction decoder for executing a designated arithmetic operation for a received data, said arithmetic operation unit outputting not only the result of the designated arithmetic operation, but also a sign information and an overflow/underflow information of the result of the designated arithmetic operation, and a saturation detecting circuit receiving the sign information and the overflow/underflow information for controlling a selector in such a manner that if an overflow has occurred when the sign information indicates the positive, said selector selects a positive maximum value; if an underflow has occurred when the sign information indicates the negative, said selector selects a negative maximum value; and if neither the overflow nor the underflow has occurred, said selector selects the result of arithmetic operation outputted from said arithmetic operation unit. said saturation detecting circuit operating, if the overflow has occurred when the sign information indicates the positive, or if the underflow has occurred when the sign information indicates the negative. to set a saturation flag of a PSW indicative of existence/non-existence of a saturation processing, and said saturation flag being reset by only execution of a data transfer instruction by said arithmetic operation unit, after said saturation flag is set.

6. A data processing system claimed in Claim 5 wherein said saturation flag is formed of a RS flipflop having a reset input activated by execution of said data transfer instruction.

FIGURE 1 PRIOR ART

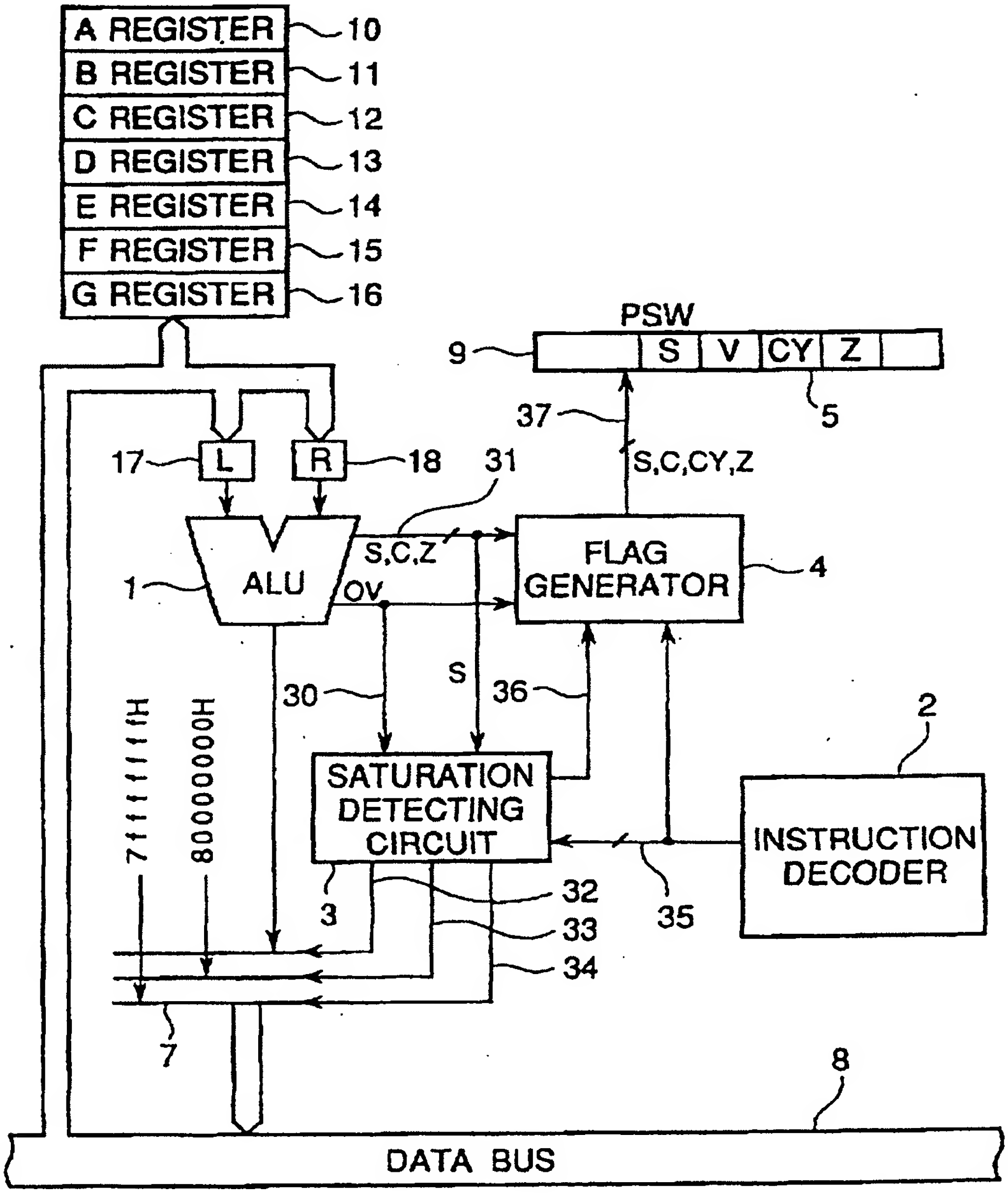


FIGURE 2 PRIOR ART

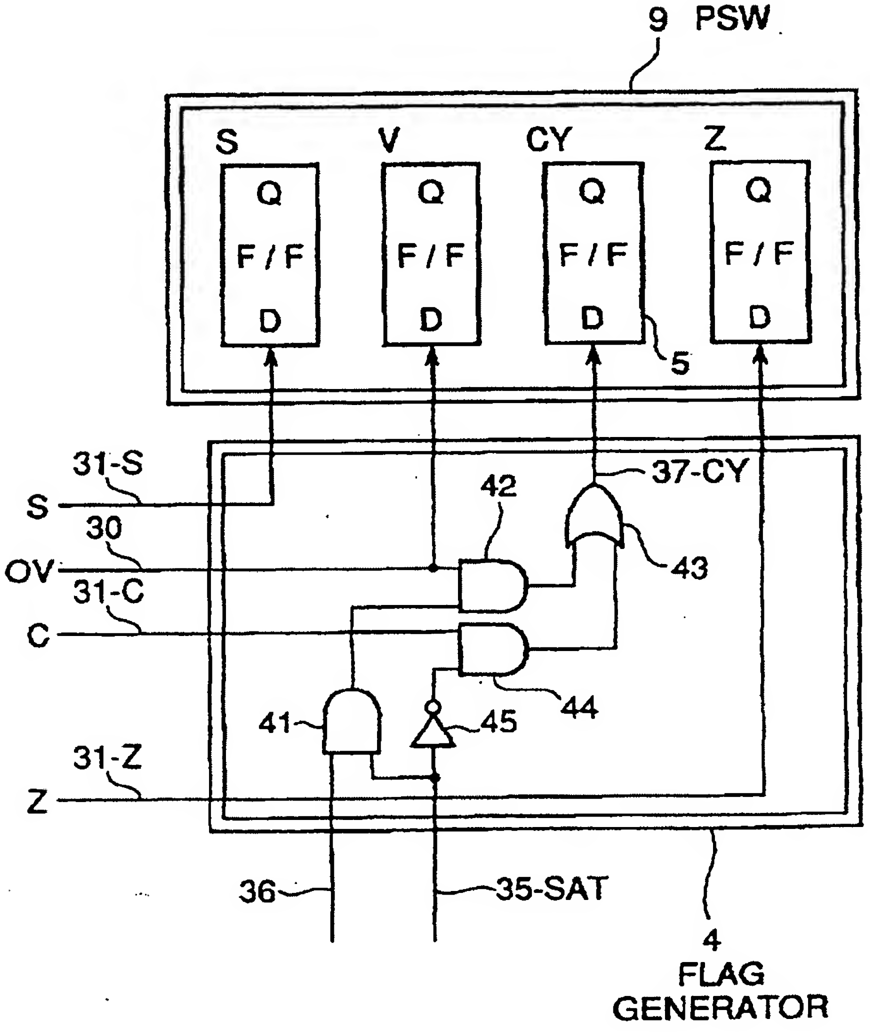


FIGURE 3 PRIOR ART

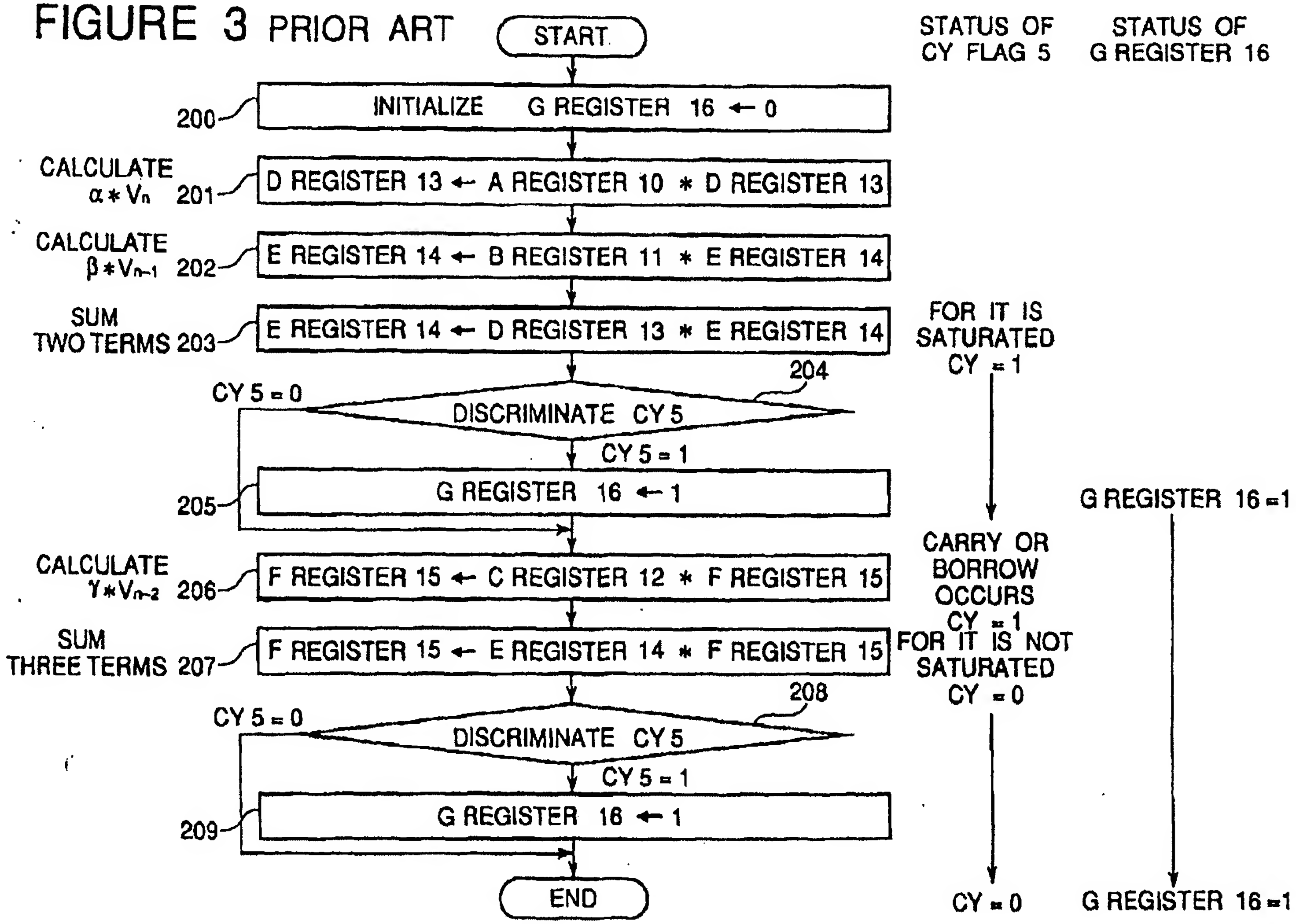

START
STATUS OF CY FLAG 5
STATUS OF G REGISTER 16
200
INITIALIZE G REGISTER 16 ← 0
CALCULATE $\alpha * V_n$ 201
D REGISTER 13 ← A REGISTER 10 * D REGISTER 13
CALCULATE $\beta * V_{n-1}$ 202
E REGISTER 14 ← B REGISTER 11 * E REGISTER 14
SUM TWO TERMS 203
E REGISTER 14 ← D REGISTER 13 * E REGISTER 14
FOR IT IS SATURATED CY = 1
204
CY 5 = 0
DISCRIMINATE CY 5
CY 5 = 1
205
G REGISTER 16 ← 1
G REGISTER 16 = 1
CARRY OR BORROW OCCURS CY = 1
CALCULATE $\gamma * V_{n-2}$ 206
F REGISTER 15 ← C REGISTER 12 * F REGISTER 15
SUM THREE TERMS 207
F REGISTER 15 ← E REGISTER 14 * F REGISTER 15
FOR IT IS NOT SATURATED CY = 0
208
CY 5 = 0
DISCRIMINATE CY 5
CY 5 = 1
209
G REGISTER 16 ← 1
END
CY = 0
G REGISTER 16 = 1

FIGURE 4

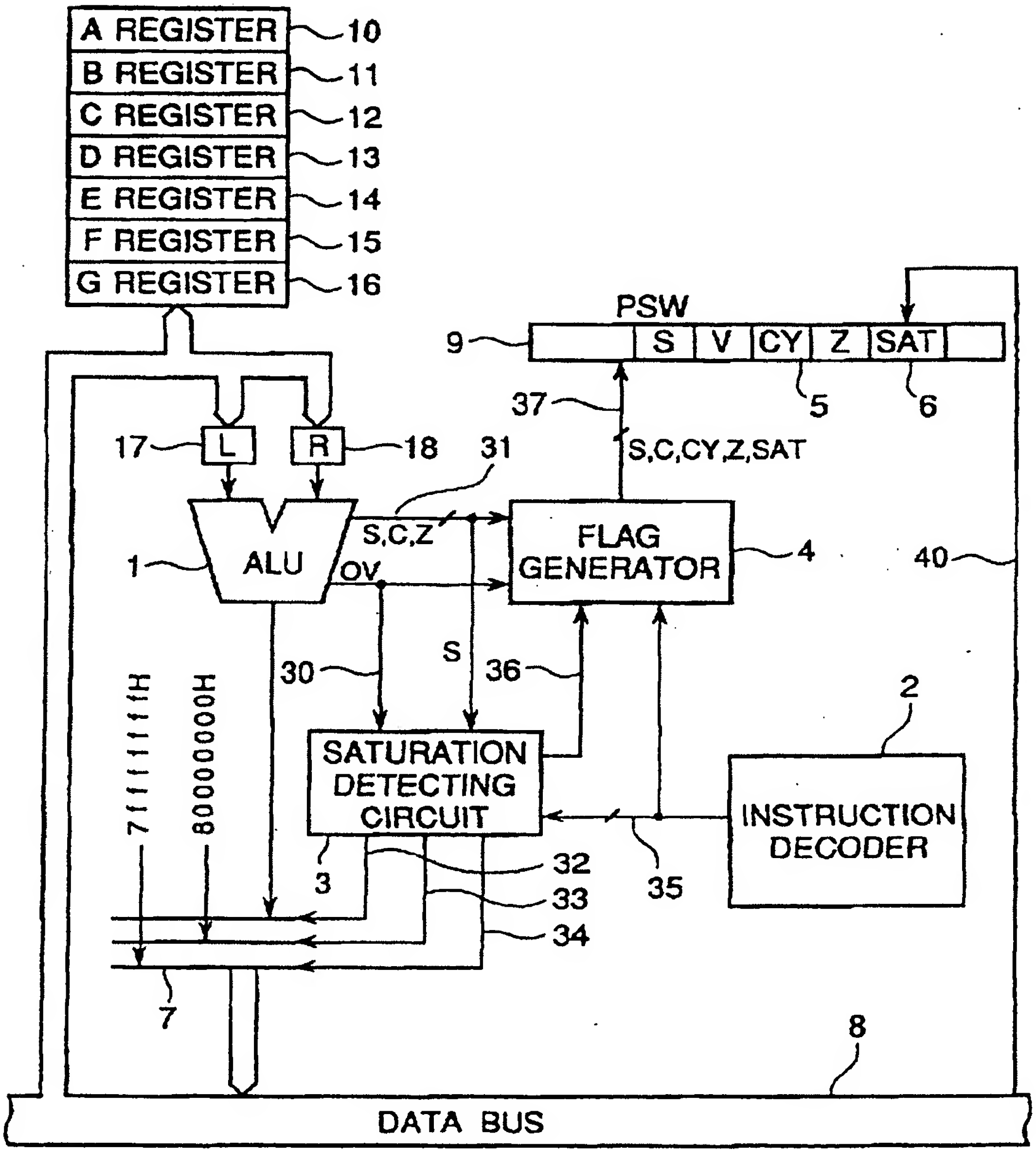

A REGISTER
B REGISTER
C REGISTER
D REGISTER
E REGISTER
F REGISTER
G REGISTER
10
11
12
13
14
15
16
PSW
9
S
V
CY
Z
SAT
5
6
37
S,C,CY,Z,SAT
17
L
R
18
31
1
ALU
S,C,Z
OV
FLAG GENERATOR
4
40
30
S
36
2
7FFFFFFFH
80000000H
SATURATION DETECTING CIRCUIT
INSTRUCTION DECODER
3
32
33
34
35
7
8
DATA BUS

FIGURE 5

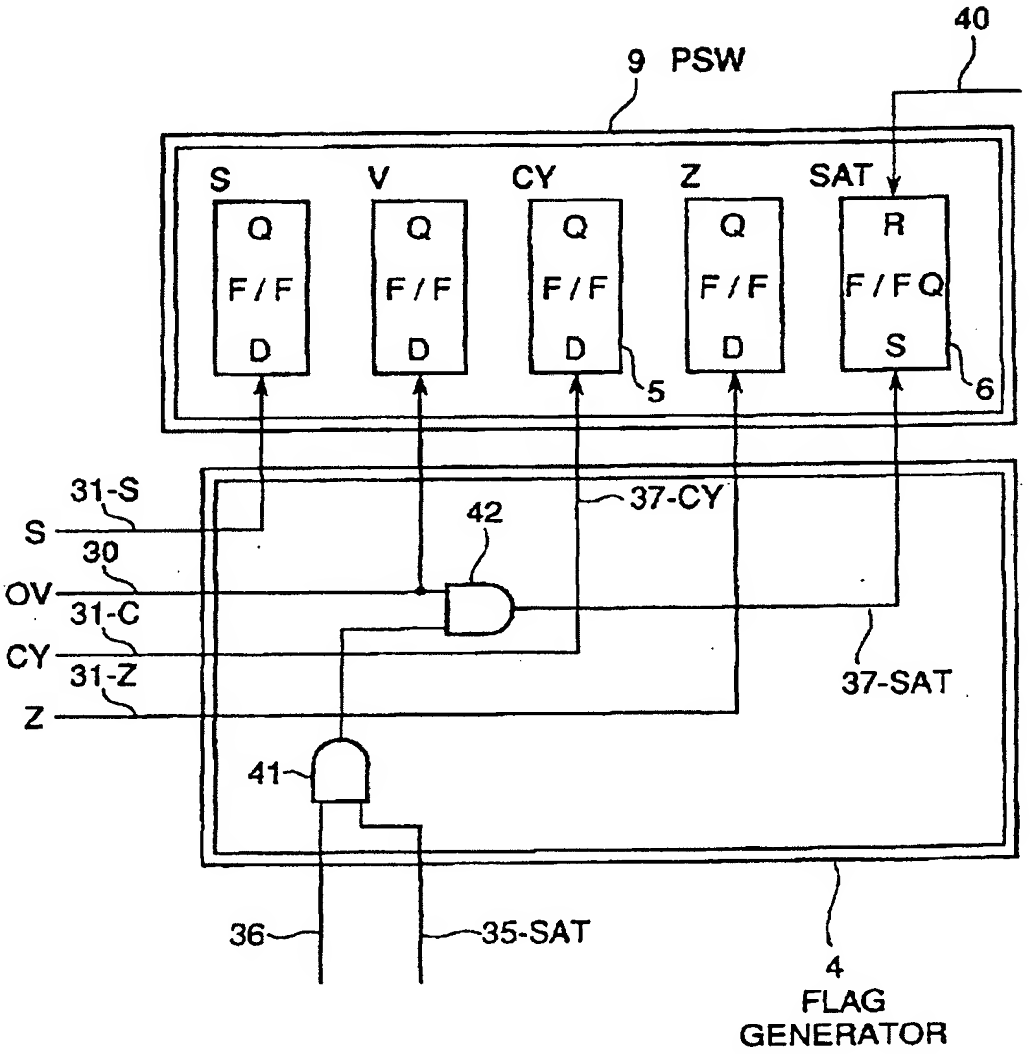
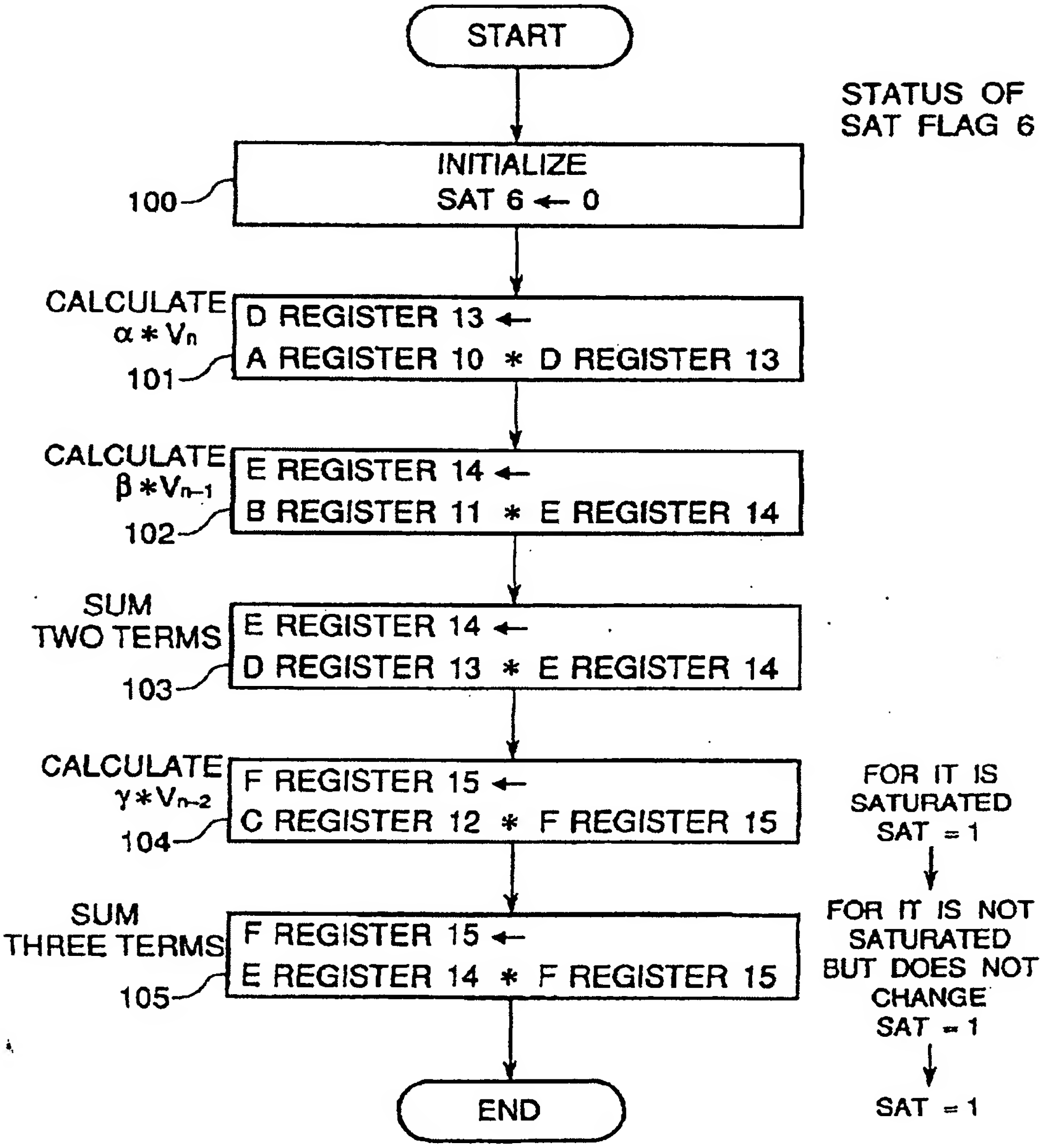

40
9 PSW
S
V
CY
Z
SAT
Q
F / F
D
R
F / F Q
S
5
6
31-S
S
30
OV
31-C
CY
31-Z
Z
42
37-CY
37-SAT
41
36
35-SAT
4
FLAG
GENERATOR

FIGURE 6


START
STATUS OF SAT FLAG 6
INITIALIZE
SAT 6 ← 0
100
CALCULATE $\alpha * V_n$
D REGISTER 13 ←
A REGISTER 10 * D REGISTER 13
101
CALCULATE $\beta * V_{n-1}$
E REGISTER 14 ←
B REGISTER 11 * E REGISTER 14
102
SUM TWO TERMS
E REGISTER 14 ←
D REGISTER 13 * E REGISTER 14
103
CALCULATE $\gamma * V_{n-2}$
F REGISTER 15 ←
C REGISTER 12 * F REGISTER 15
104
FOR IT IS SATURATED SAT = 1
SUM THREE TERMS
F REGISTER 15 ←
E REGISTER 14 * F REGISTER 15
105
FOR IT IS NOT SATURATED BUT DOES NOT CHANGE SAT = 1
END
SAT = 1

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 10 9062

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X Y	EP-A-0 130 377 (IBM) * page 37, line 7 - line 35; figure 3 * * page 43, line 7 - line 25 * ---	1-4 5,6	G06F7/48 G06F9/32
D,Y	PATENT ABSTRACTS OF JAPAN vol. 7 no. 224 (P-227) ,5 October 1983 & JP-A-58 115544 (MITSUBISHI) * abstract * ---	5,6	
A	EDN ELECTRICAL DESIGN NEWS, vol. 28, no. 1, January 1983 NEWTON, MASSACHUSETTS US, pages 133-135, GUTTAG 'Buil-in overflow detection etc...' * page 135, column 1 * ---	1	
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 24, no. 7a, December 1981 NEW YORK US, pages 3126-3127, 'Fixed point overflow exception detection' * paragraph 2 - paragraph 5 * -----	1	TECHNICAL FIELDS SEARCHED (Int.Cl.6) G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	30 August 1995	Cohen, B

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document